

Trabajo Fin de Grado

APLICACIÓN PARA LA GENERACIÓN DE RECORRIDOS DE ORIENTACIÓN EN PARQUES EN EL AMBITO EDUCATIVO

Autor

Sandra Pérez Díaz

Director

José García Moros

Escuela de Ingeniería y Arquitectura
Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Septiembre 2015

Agradecimientos

En primer lugar quisiera agradecer a José García Moros, mi director, por brindarme la oportunidad de realizar este trabajo, así como por su apoyo y ayuda a lo largo del desarrollo del mismo.

También quisiera acordarme de mi familia y amigos por apoyarme en todo momento desde que empecé la carrera. En especial a ti, por hacer de mi lo que soy hoy.

Y por supuesto a toda esa familia “esmocheda” que me ha acompañado todos estos años, y que está presente día a día. Sin duda lo mejor que me ha regalado esta carrera sois vosotros. Gracias de corazón, a todos.

APLICACIÓN PARA LA GENERACIÓN DE RECORRIDOS DE ORIENTACIÓN EN PARQUES EN EL ÁMBITO EDUCATIVO

RESUMEN

Este Trabajo Fin de Grado tiene como finalidad el desarrollo y programación de una aplicación que será ejecutada sobre dispositivos tipo tablet bajo la plataforma Phonegap y será utilizada por profesores de Educación Física en actividades educativas relacionadas con la orientación. Gracias a la utilización de Phonegap se permite adaptar el código de la aplicación a distintos sistemas operativos existentes en el mercado (iOS, Android, Windows Phone...).

Por tanto, esta aplicación propone una solución para el personal docente de Educación Física brindando la posibilidad de generar de manera dinámica e informatizada los distintos recorridos empleados en las actividades de orientación desarrolladas con sus alumnos.

Proporciona la posibilidad de posicionar distintas balizas sobre mapas y así mismo generar diferentes recorridos a partir de ellas. La aplicación también permite asignar una pequeña descripción y/o fotografía(s) a cada baliza. Además incluye la opción de generar un documento con la información del recorrido y de todas las balizas que lo componen.

Para ello se ha utilizado la API de Google Maps en todo momento, tanto para la visualización del mapa como para el posicionamiento de balizas. Además, toda la información generada se almacena de forma dinámica en una base de datos, alojada en un servidor externo, utilizando MySQL, para acceder a ella de forma rápida y sencilla.

Índice general

1. Introducción	1
1.1 Motivación y objetivos.....	2
1.2 Estado del arte	2
1.3 Organización de la memoria	3
1.4 Cronograma.....	4
2. Descripción del sistema.....	5
2.1 Visión general.....	5
2.2 Público objetivo.....	8
2.3 Requisitos fundamentales.....	8
2.4 Arquitectura propuesta.....	9
3. Descripción de la aplicación	13
3.1 Información general	13
3.2 Funcionamiento de la aplicación	16
3.2.1 Pantalla Inicio	17
3.2.2 Pantalla Principal	18
3.2.2.1 Menú Balizas	20
3.2.2.2 Menú Recorridos	22
3.2.2.3 Cambiar estilo Mapa	28
3.2.2.4 Guardar y salir	28
3.2.2.5 Modo parque y botón Home.....	29
3.2.3 Pantalla Configuración de Baliza.....	29
3.2.4 Pantallas Ayuda y no Conexión	30
4. Descripción del servidor	31
4.1 Planteamiento general	31
4.2 Funcionamiento del servidor	32
4.3 Base de datos	32
4.3.1 Tabla Balizas.....	32
4.3.2 Tabla Recorridos	33
4.3.3 Tabla Usuarios.....	33
5. Pruebas y evaluación.....	34

5.1 Pruebas durante y tras el desarrollo	34
6. Conclusiones y líneas futuras	36
6.1 Opinión personal.....	36
6.2 Conclusiones	36
6.3 Líneas futuras	37
7. Bibliografía	38
A. Bases de datos	42
A.1 Introducción	42
A.2 Base de datos del dispositivo	42
A.3 Base de datos del servidor.....	44
B. Elección de plataforma	49
C. Manual de usuario	51
C.1 Introducción.....	51
C.2 Cómo utilizar la aplicación	51

Índice de figuras

Figura 1.1 Diagrama de Gant	4
Figura 2.1 Esquema general del sistema	6
Figura 2.2 Aspecto de una baliza de una carrera de orientación, con identificación U-2	6
Figura 2.3 Aspecto general de un recorrido.....	7
Figura 2.4 Esquema general de la arquitectura elegida.....	12
Figura 3.1 Diagrama UML del elemento baliza	14
Figura 3.2 Diagrama UML del elemento recorrido	15
Figura 3.3 Interacción entre las pantallas de la aplicación	16
Figura 3.4 Diagrama de navegación de la pantalla Inicio.....	17
Figura 3.5 Diagrama de navegación de la pantalla principal.....	18
Figura 3.6 Apariencia inicial de la página principal	18
Figura 3.7 Formato de la pantalla principal.....	19
Figura 3.8 Apariencia del Menú Balizas.....	20
Figura 3.9 Diagrama navegación del Menú Balizas	20
Figura 3.10 Apariencia de una baliza posicionada sobre el mapa	21
Figura 3.11 Proceso Cargar Balizas	22
Figura 3.12 Apariencia del Menú Recorridos	23
Figura 3.13 Diagrama navegación del Menú Recorridos.....	23
Figura 3.14 Ilustración proceso división coordenadas	24
Figura 3.15 Funciones adicionales de los recorridos.....	26
Figura 3.16 Recorrido modificado al añadir 2 balizas.....	27
Figura A.1 Estructura de la tabla de la base de datos del dispositivo.....	43
Figura A.2 Estructura tabla balizas (BBDD externa).....	44

Figura A.3 Estructura tabla recorridos (BBDD externa)	46
Figura A.4 Estructura tabla usuarios (BBDD externa)	47
Figura B.1 Cuadro resumen del reparto de las versiones de Android en Agosto de 2015	50
Figura C.1 Pantalla inicio y botón de registro	52
Figura C.2 Campos a rellenar para registro.....	52
Figura C.3 Campos a rellenar para inicio de sesión	52
Figura C.4 Pantalla principal de la aplicación.....	53
Figura C.5 Pantalla buscar mapa por dirección.....	53
Figura C.6 Pantalla mapa seleccionado y menús	54
Figura C.7 Pantalla lista de mapas modificados	54
Figura C.8 Pantalla de configuración de una baliza	55
Figura C.9 Opciones de borrado de una baliza.....	55
Figura C.10 Distintos tipos de mapas posibles	56
Figura C.11 Balizas del recorrido (1,2 y 3) y cuadro para guardar recorrido.....	56
Figura C.12 Cuadro de recorridos existentes e información recorrido seleccionado.....	57
Figura C.13 Cuadro informativo sobre baliza en pantalla recorrido.....	58
Figura C.14 Formato informe completo.....	59
Figura C.15 Formato informe compacto.....	59
Figura C.16 Pantalla ayuda.....	60
Figura C.17 Pantalla de no conexión	61

Capítulo 1

Introducción

En la actualidad los dispositivos electrónicos, como los smartphones o tablets, están cada vez más introducidos en la vida cotidiana de las personas. Gracias a la utilidad que proporcionan estos dispositivos, tanto a nivel personal como profesional, se han ido abriendo paso en los centros docentes como una forma de apoyo tanto para profesores como para alumnos. Por ello, se encuentran cada vez más presentes en los centros educativos, tanto es así que hoy en día la mayoría de colegios cuentan con pizarras electrónicas que pueden ser manejadas desde una tablet u ordenador.

Por otro lado, el fomento de la actividad física es muy importante, tanto a nivel educativo como a nivel de salud. En el campo educativo, son los profesores de Educación Física quienes se encargan de esta materia. Existe una actividad muy dinámica y divertida para los alumnos realizada en estas clases, el deporte de orientación [1]. Ésta es una forma peculiar de entender la actividad física y el deporte en el medio natural. Básicamente consiste en realizar una carrera campo a través, pero completando un recorrido en el menor tiempo posible, pasando por una serie de puntos o balizas localizados sobre el mapa y siguiendo un orden impuesto.

Con el desarrollo de la aplicación que propone este trabajo se pretende en todo momento facilitar la tarea de creación de estos recorridos al profesor, tarea que actualmente debe desarrollar de forma manual sobre un mapa en papel. Evidentemente, tras el desarrollo del recorrido virtual, el profesor deberá ir a colocar las distintas balizas del mapa en la posición concreta.

Con el objetivo de que esta aplicación fuese lo más útil y usable posible se ha contado con el asesoramiento de personal del ámbito de la orientación como la Asociación PRAMES [2], el club IBON [3] o el grupo EFYPAF [4].

1.1 Motivación y objetivos

La iniciativa de este Trabajo Fin de Grado nace a partir de la colaboración del grupo de investigación “Communications Networks and Information Technologies for Health and Quality of experience”(CeNITEQ)[5] con el grupo de investigación “Educación Física y Promoción de la Actividad Física” (EFYPAF), un grupo de la Universidad de Zaragoza cuyo objetivo es fomentar la actividad física desde los centros educativos.

Tras varias reuniones, la idea conjunta propuesta fue la de crear una aplicación que facilitase el trabajo previo que debe realizar un profesor de Educación Física en las tareas que desarrolla con sus alumnos sobre orientación. El objetivo del TFG es por tanto, generar una aplicación de uso sencillo, gratuita y ejecutada sobre un dispositivo tablet, independientemente de su sistema operativo, que permitiese al personal docente posicionar balizas o puntos de control y a partir de ellos, generar recorridos sobre diferentes mapas.

Exactamente, tras la realización del trabajo, se perseguía obtener una aplicación que cumpliese con los siguientes requerimientos:

- Mostrar distintos mapas buscando por dirección.
- Colocar/borrar balizas sobre el mapa.
- Seleccionar distintas balizas para generar diferentes recorridos.
- Introducir información y/o fotografías ilustrativas de cada baliza.
- Generar un documento con formato PDF que recoja la información del recorrido generado, tanto para el profesor como para el alumno.

La aplicación originada tiene objetivo didáctico ya que los profesores de Educación Física la podrán emplear para crear unidades didácticas de orientación en su asignatura. Sin embargo, la aplicación no limita su uso a profesores si no que podría ser utilizada como recurso para generar rutas turísticas. Por tanto en el enfoque no debe dejarse de lado la funcionalidad y usabilidad de la aplicación, ya que ésta podría ser utilizada por cualquier usuario, no sólo por personal docente.

1.2 Estado del arte

Previamente al desarrollo de la aplicación, se analizaron los directos competidores de la misma, es decir, las alternativas que existían en ese momento en el mercado. Todas las aplicaciones encontradas son de pago y son ampliamente utilizadas en el mundo de la orientación. A modo de resumen se exponen las características de estas herramientas:

- **SportIdent:** Software empleado en carreras de orientación oficiales. Esta herramienta consta de unas pequeñas tarjetas que los participantes llevan en el dedo y debe ser introducido en los distintos puestos de control. Con ello, se permite a los participantes registrar su actividad a su paso por las distintas balizas e imprimir dicha información al

terminar el recorrido.

- **OCAD:** Software con licencia empleado en la cartografía oficial de mapas. Permite al cartógrafo diseñar mapas con diferentes niveles de complejidad y detalle. Es una herramienta muy útil que cubre las necesidades de los cartógrafos, pero a su vez es bastante compleja pues se necesitan amplios conocimientos sobre cartografía.
- **RouteGadget:** Software utilizado para realizar recorridos sobre un mapa. Esta herramienta podría ser complementaria a la anterior ya que, tras el diseño del mapa, permite generar distintos recorridos sobre el mismo.

A modo de conclusión cabe decir que existen varias aplicaciones en este campo, pero todas ellas presentan las mismas carencias respecto a la propuesta hecha en este TFG. Por un lado, son herramientas muy profesionales y esto se traduce en una elevada complejidad de uso. Y por otro lado, su precio de mercado es alto lo que también supone una limitación. Por ello, nuestra propuesta es una plataforma que rompe tanto con la limitación de uso debido a la complejidad, como con la del precio de mercado. Es decir, se propone una aplicación gratuita, con funciones básicas pero completas y capaz de ser utilizada por todo tipo de usuarios.

1.3 Organización de la memoria

El contenido de la memoria se estructura de la siguiente manera:

- En el **Capítulo 1** se realiza una breve introducción al TFG, un pequeño análisis del estado actual del mercado y una exposición del objetivo final que se persigue con su desarrollo.
- En el **Capítulo 2** se explica el sistema, haciendo un análisis del público objetivo, detallando los requisitos y especificaciones de la aplicación y presentando la arquitectura que sigue la misma.
- En el **Capítulo 3** se detalla el desarrollo de la aplicación, la estructura de menús, el funcionamiento.
- En el **Capítulo 4** se detalla la información relativa al servidor, el funcionamiento del mismo y la estructura de las bases de datos.
- En el **Capítulo 5** se explican las fases de pruebas a las que se ha sometido la aplicación tanto durante el proceso de programación como tras el mismo.
- En el **Capítulo 6** se recoge un análisis de las líneas futuras que podría tener la aplicación, así como las conclusiones y una breve opinión personal.

Además, en la parte final del trabajo se pueden encontrar varios Anexos que complementan la información expuesta en la memoria:

- En el **Anexo A** se explican los detalles sobre las bases de datos, tanto interna como externa.
- En el **Anexo B** se exponen los aspectos característicos y motivos de elección de la plataforma utilizada.
- En el **Anexo C** se detalla un pequeño manual de usuario que permita comprender el uso de la aplicación.

1.4 Cronograma

En la Figura 1.1 se muestra el reparto del tiempo invertido en las diferentes tareas en que se ha dividido el desarrollo de la aplicación, separadas en acotación de la temática del trabajo, aprendizaje de conocimientos de programación (HTML, JavaScript, PHP, jQuery...), análisis del sistema, implementación de la aplicación, fase de pruebas de funcionamiento y finalmente la redacción de la memoria.

Como se puede observar en la figura, se utilizaron los primeros meses para refrescar conocimientos de programación, aunque esta fase se ha solapado con todas las demás, ya que siempre ha sido necesario aprender conceptos de programación para avanzar y completar el desarrollo del trabajo. De la misma manera, la fase de pruebas se ha llevado a cabo simultáneamente a la fase de implementación, ya que ha sido necesario comprobar el correcto funcionamiento del código al mismo tiempo que se iba elaborando.

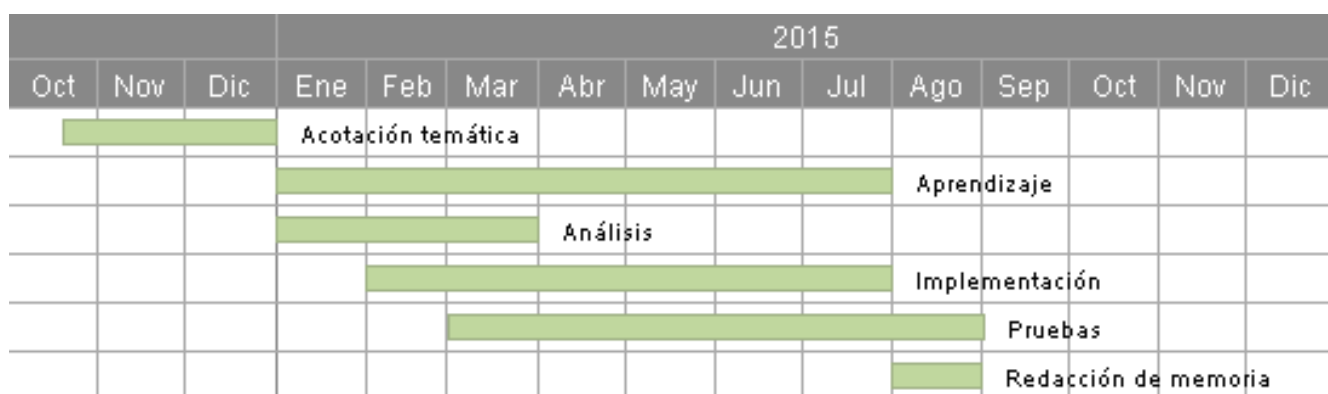


Figura 1.1 Diagrama de Gant

Capítulo 2

Descripción del sistema

En este capítulo se muestra una pequeña explicación de los elementos fundamentales que intervienen en las carreras de orientación: por un lado las balizas, que son los puntos de control que se posicionarán sobre el mapa y por otro los recorridos, que están compuestos por un conjunto ordenado de balizas. Además se hace un pequeño análisis sobre el público al que va dirigida la aplicación. Y por último se enumeran los requisitos fundamentales que nuestra aplicación debe contemplar y se hace un primer análisis sobre la arquitectura propuesta para nuestro sistema.

2.1 Visión general

Nuestro sistema está basado en la utilización de dos elementos fundamentales, balizas y recorridos. Estos tienen en común un tercer elemento, el mapa, sobre el que se posicionan las balizas y se generan los recorridos. En la Figura 2.1 se ilustra la relación existente entre ellos. Para comprender la estructura y relación entre los elementos de este sistema, hemos de explicar por separado los conceptos fundamentales que definen un recorrido de orientación:

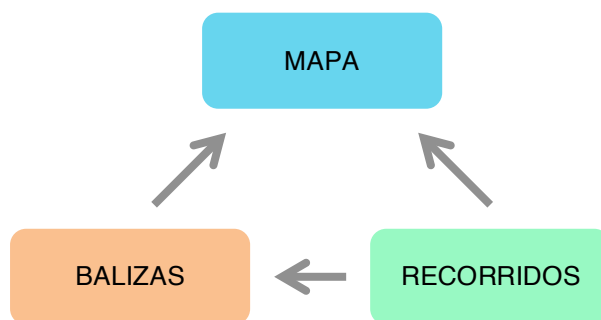


Figura 2.1 Esquema general del sistema

- **Balizas:**

Son los puntos clave que se posicionan sobre el mapa y permiten construir un recorrido. En una carrera de orientación son los puntos de control por los que debe pasar el corredor para terminar la carrera con éxito. En estos puntos, que generalmente están identificados con un número y/o una letra, se debe llevar a cabo una “tarea” que indique que el corredor ha pasado por ahí.

En la aplicación planteada, estos puntos de control pueden ser posicionados por el usuario en cualquier posición del mapa que deseemos. Esto se realiza así, en contrapartida a lo que inicialmente se planteaba, trabajar con un registro de todas las balizas fijas que actualmente hay por los parques. Esta idea se descartó debido a que estas balizas a menudo son manipuladas y/o destruidas por individuos que nada tienen que ver con la orientación, ya que están expuestas en todo momento al público y generalmente no se realiza un registro exhaustivo de las balizas que cada entidad coloca.

En una carrera de orientación suelen tener el aspecto reflejado en la Figura 2.2, pero a menudo podemos ver balizas con aspectos de lo más variado, desde un trozo de pintura en una piedra hasta los mismos elementos del parque podrían actuar de balizas. Esto además de permitir desarrollar distintos tipos de recorridos (botánico, histórico, etc...) permite al profesor diseñar recorridos a su gusto.



Figura 2.2 Aspecto de una baliza de una carrera de orientación, con identificación U-2

Cabe destacar que sobre un mismo mapa pueden ser posicionadas tantas balizas como se desee, independientemente de si forman parte o no de un recorrido.

- **Recorridos:**

Están formados por un conjunto ordenado de balizas que componen la carrera en su totalidad. Pueden tener, como decimos, muchas temáticas en función de las balizas que lo componen. El objetivo de las carreras de orientación es conseguir completar el recorrido generalmente en el menor tiempo posible, aunque existen recorridos en los que el tiempo no es partícipe. En la Figura 2.3 podemos observar el aspecto general de un recorrido típico.

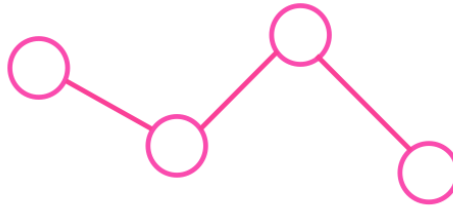


Figura 2.3 Aspecto general de un recorrido.

Con la aplicación propuesta, se pueden realizar tantos recorridos como se deseen sobre un mismo mapa, pero siempre utilizando balizas previamente posicionadas sobre el mapa.

- **Mapas:**

Es el tercer elemento que compone nuestro sistema. Aunque ciertamente no es un elemento como tal, puesto que es inherente a cada baliza y recorrido, es importante tenerlo presente y aclarar que conforma un punto de unión entre balizas y recorridos. Como se ha comentado, tanto las balizas como los recorridos son posicionadas o generados sobre un mapa. Sin embargo, a diferencia de los otros elementos, éste no dispone de una tabla propia en la base de datos (BBDD) externa sino que únicamente será un campo en las tablas de los otros elementos.

Por tanto, es fundamental entender que en nuestro sistema, balizas y recorridos están completamente separados pero dependen totalmente el uno del otro. Es decir:

- Se necesita posicionar alguna baliza sobre un mapa para llevar a cabo el diseño de un recorrido sobre ese mapa.
- Pero un recorrido puede no contener todas las balizas que existen sobre el mapa.

Tras este pequeño análisis del sistema, necesario para entender posteriormente la estructura y funcionamiento de la aplicación, podemos abordar temas sobre la aplicación como los requisitos y/o la arquitectura elegida.

2.2 Público objetivo

La plataforma planteada se ha diseñado pensando en un grupo de población específico, aquellos profesores de Educación Física que desarrollen actividades de orientación con sus alumnos. Estas actividades se pueden llevar a cabo tanto con niños como con personas adultas, luego es importante contemplar que el rango de edad de utilización será amplio.

Esto debe ser así porque aunque bien es cierto que la aplicación en sí misma está pensada para ser utilizada sólo por el profesor de la actividad, la plataforma proporciona un documento con información sobre el recorrido que deberá ser sencillo y claro, pues debe ser comprendido por todos los alumnos, bien sean niños o adultos.

Por ello, tanto para la apariencia de menús de la aplicación como para el documento .pdf que proporciona, se persigue un estilo sencillo pero completo e intuitivo que permita identificar de manera clara los puntos clave en el mapa.

2.3 Requisitos fundamentales

Tras una primera visión general y un pequeño análisis del público objetivo es preciso definir y detallar los requisitos funcionales que caracterizarán nuestra aplicación:

- **RF1:** La aplicación debe ser capaz de diferenciar entre usuarios y mostrar únicamente aquellos mapas/balizas/recorridos que pertenezcan a ese usuario.
- **RF2:** La aplicación debe permitir al usuario la correcta visualización de un mapa sobre el que trabajar.
- **RF3:** La aplicación debe aceptar la ubicación de balizas en diferentes posiciones del mapa.
- **RF4:** La aplicación debe soportar la creación de distintos recorridos sobre un mismo mapa a partir de balizas ya incluidas en él.
- **RF5:** La aplicación debe permitir al usuario la posible eliminación de balizas y/o recorridos ya creados sobre un mapa.
- **RF6:** La aplicación debe admitir la modificación de los campos con información de cada baliza.
- **RF7:** La aplicación aceptará la realización de dos fotografías por cada baliza que serán almacenadas en un servidor externo alojado en un dominio web y consultarlas cuando sea necesario.

- **RF8:** La aplicación debe permitir la modificación de los distintos recorridos. Tanto si esta modificación implica eliminar balizas existentes en el recorrido o añadir nuevas balizas al recorrido.
- **RF9:** La aplicación permitirá mostrar las balizas creadas anteriormente por el usuario sobre el mapa seleccionado.
- **RF10:** La aplicación permitirá mostrar en una tabla los distintos recorridos generados por el usuario sobre el mapa seleccionado.
- **RF11:** La aplicación debe contabilizar la distancia en metros de cada recorrido, y permitirá introducir manualmente un tiempo aproximado de realización a cada recorrido.
- **RF12:** La aplicación será capaz de comunicarse con un servidor PHP alojado en un dominio web y diferenciar el tipo de petición.
- **RF13:** La aplicación debe almacenar la información sobre las balizas y los recorridos y los usuarios en una base de datos externa. Y debe permitir la lectura de la información y presentarla correctamente sobre el mapa.
- **RF14:** La aplicación debe ser capaz de mostrar la información (mapa, balizas y recorrido) que se desea imprimir, y generar un documento .pdf a partir de la misma.
- **RF15:** La aplicación debe distinguir entre dos tipos de destinatarios del documento .pdf, alumno y profesor, con diferentes características. También permite generar tres tipos de informes para el profesor: completo, compacto con fotos y compacto sin fotos.
- **RF16:** La aplicación debe ser capaz de enviar por correo electrónico al profesor y el documento generado con información sobre el recorrido.
- **RF17:** La aplicación permite en todo momento conocer el estado de la conexión a internet y no permite su uso mientras no exista conexión.

2.4 Arquitectura propuesta

En cuanto a la arquitectura propuesta es importante destacar los elementos básicos que la componen: un dispositivo tablet sobre el que se ejecuta la aplicación basada en Phonegap [6], un servidor y una base de datos externa alojados en un dominio web, y por último una base de datos interna alojada en el dispositivo tablet.

Para una mayor comprensión de esta arquitectura se van a describir sus elementos de manera individual. Además se va a distinguir entre los elementos que interaccionan entre ellos, y las herramientas utilizadas en la interacción.

En cuanto a los elementos fundamentales, encontramos la tablet y el servidor, y cada uno de ellos dispone además de una base de datos:

- **Tablet:** este elemento es fundamental, pues será el dispositivo que se utilice para ejecutar la aplicación. No importa que sistema operativo utilice, *iOs*, *Android*, *Blackberry*, etc..., dado que se ha programado utilizando Phonegap, que será explicado posteriormente con mayor detalle.
- **Base de Datos Interna:** es la base de datos alojada en el dispositivo tablet. Su presencia es necesaria para proporcionar un correcto funcionamiento del acceso de los usuarios, así mismo almacena información que permite evitar el Inicio de Sesión en cada ejecución o información para deshacer los cambios no guardados. En el Anexo A se muestra una tabla explicativa de esta BBDD.
- **Servidor externo:** es imprescindible la presencia de un servidor externo tanto para almacenar la información en una base de datos externa como para guardar las fotografías de las balizas, puesto que alojar directamente la fotografía en la base de datos suponía un exceso elevado de memoria.
- **Base de Datos Externa:** es la base de datos alojada en el servidor externo y será aquí donde se almacene todo tipo de información. En nuestro caso serán necesarias tres tablas, una para guardar todas las balizas que se posicionen, otra para todos los recorridos que se generen y una tercera para el almacenamiento de los usuarios. En el Anexo A se muestran las distintas tablas explicativas de esta BBDD.

Y en cuanto a las herramientas utilizadas para la interacción entre estos elementos, se encuentran los distintos recursos necesarios para el desarrollo de la aplicación, la plataforma Phonegap, y distintos plugins y librerías:

- **SDK Eclipse [7]:** Entorno de trabajo para la programación de la aplicación. Lugar donde se lleva a cabo la programación de todos los componentes de la aplicación. Será aquí donde creemos un proyecto Phonegap.
- **Phonegap:** Es la plataforma encargada de adecuar el código de la aplicación, programado en *HTML5*, *JavaScript* y *CSS3*, para su correcto funcionamiento en cualquier plataforma (iOs, Android...). Phonegap dispone de unos plugins que permiten a la aplicación acceder a las distintas características de las que una aplicación nativa dispondría.
- **Plugins:** Son complementos aportan funciones nuevas y generalmente muy específicas a la aplicación. Éstos son propios de Phonegap y permiten la interacción con las funciones nativas del sistema operativo. En cuanto a los plugins utilizados, destacar cuatro:
 - Cámara: permite el acceso tanto a la cámara frontal como externa de la tablet. Este plugin es importante a la hora de realizar las fotos correspondientes de cada baliza.

- Red: permite conocer en todo momento el estado de la conexión a internet. Es muy importante puesto que si no se dispone de conexión la aplicación no permite su funcionamiento.
- SQLite: este plugin permite crear/leer/escribir sobre una base de datos interna.
- Geolocalización: permite conocer la localización GPS del dispositivo.
- **Librerías externas**: Son conjuntos de funciones que nos permiten dotar a nuestra aplicación de nuevas funciones. En nuestra aplicación hemos utilizado:
 - API de Google Maps [8]: son el conjunto de librerías mas importante de la aplicación ya que son utilizadas para la visualización de los mapas y la interacción con ellos. Para su utilización se requiere de código JavaScript.
 - jQuery: biblioteca de JavaScript que, entre otras cosas, permite interacción con la técnica *AJAX*.
 - HTML2PDF: esta librería permite generar a partir de texto html un archivo .pdf. Para su utilización se requiere lenguaje *PHP*.
 - Swift Mailer: librería que permite el envío de correos electrónicos. Su utilización se lleva a cabo desde *PHP*.

Además, para la utilización de estas herramientas es necesario tener conocimientos sobre distintos lenguajes de programación y los distintos tipos de bases de datos utilizados:

- HTML (HyperText Markup Language): Es el lenguaje de programación que utiliza Phonegap.
- JavaScript: Lenguaje de programación utilizado en el lado del cliente. Con él se puede interactuar con los usuarios. Aquí es preciso destacar la utilización de *AJAX (Asynchronous JavaScript And XML)* que permite mediante lenguaje JavaScript la transferencia de información. Se ha empleado *JSON (JavaScript Object Notation)* como tipo de dato de *AJAX*.
- PHP (Hypertext Preprocessor): Lenguaje de programación que es ejecutado en el lado del servidor. Se han programado una serie de archivos en este formato que han sido almacenados en el servidor para poder interactuar con él.
- mySQL: Base de datos externa en el servidor. La información se maneja gracias a secuencias SQL. Lugar donde se almacena la información de los usuarios, mapas, balizas y recorridos.

- SQLite: Base de datos interna. Lugar donde se almacenan datos para el inicio de sesión.

Una vez tenemos descritos todos los elementos necesarios para el desarrollo de la aplicación es necesario analizar las distintas relaciones existentes entre ellos:

- Por un lado existe una relación directa entre el sistema operativo de la tablet y Phonegap. Ésta se lleva a cabo gracias a los plugins que permiten que el código generado pueda acceder a las funciones nativas del sistema operativo, en este caso la cámara, el almacenamiento interno, la conexión a internet y la geolocalización.
- Por otro lado, las bibliotecas como jQuery permiten interactuar con el servidor y mediante AJAX, código PHP y secuencias SQL solicitar y/o recibir información de la base de datos externa. La interacción con la base de datos interna se realiza gracias a secuencias SQLite.

Se puede observar de forma ilustrativa esta arquitectura en la Figura 2.4:

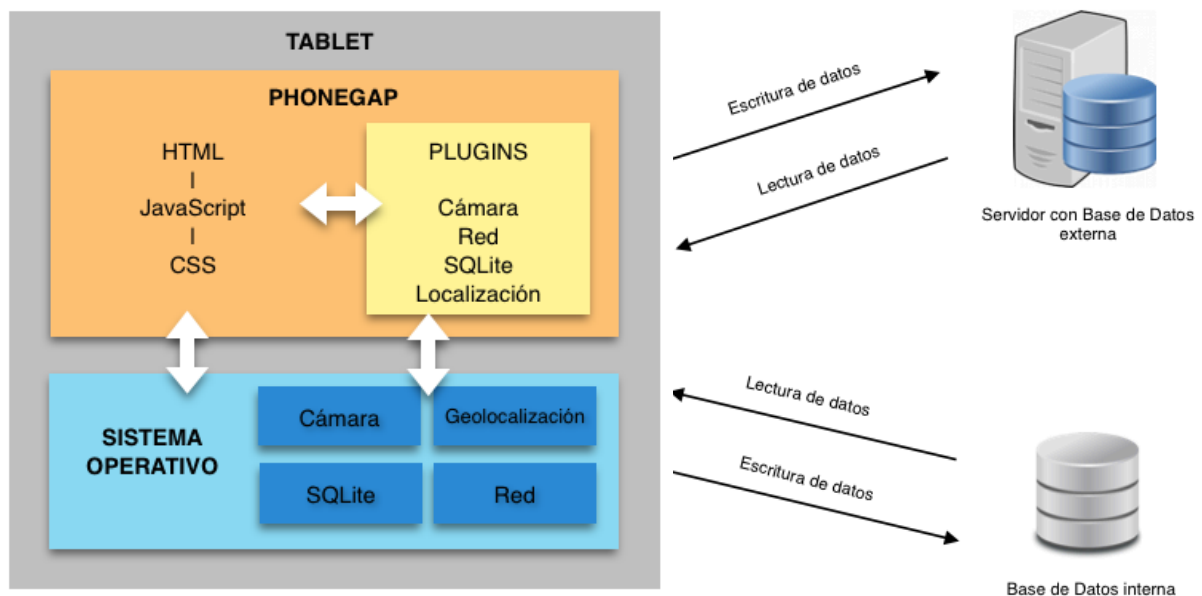


Figura 2.4 Esquema general de la arquitectura elegida

Capítulo 3

Descripción de la aplicación

En este capítulo se describe la aplicación desarrollada. En primer lugar se presenta la información que es necesario considerar, distinguiendo entre Balizas y Recorridos. Posteriormente se presenta la organización de la misma en términos de menú, pantallas y funciones que componen el funcionamiento de la aplicación.

3.1 Información general

Debido a que la aplicación está desarrollada utilizando Phonegap, no existen clases que permitan entender la estructura de la misma. No obstante, a modo de ilustración y para una mayor comprensión, se han creado dos elementos principales que son la base de la aplicación. Pese a que fueron definidos en el primer capítulo, realizaremos un pequeño recordatorio. El primero de ellos es el elemento Baliza, que corresponde con aquellos puntos de control que serán localizados sobre el mapa, mientras que el segundo es el elemento Recorrido, siendo este un conjunto ordenado de balizas y la tarea final a realizar.

Cada elemento constituye un factor fundamental en la aplicación y tal y como se ha comentado anteriormente, ambos están separados pero ampliamente relacionados. Tanto el elemento Baliza como el Recorrido tienen asociadas unas operaciones de funcionamiento que además corresponden con los menús de cada elemento en la parte visible de la aplicación. Además cada uno dispone de un conjunto distinto de parámetros que caracterizan al elemento.

El elemento Baliza se caracteriza individualmente según el mapa y el usuario al que corresponda, una identificación numérica, una pequeña descripción, las coordenadas donde se encuentra localizada, una tarea y una solución, que permitirán el control de paso por la baliza, y dos imágenes. A su vez, este elemento dispone de cuatro operaciones: añadir, borrar, configurar y cargar existentes. En la Figura 3.1 se puede observar a modo de resumen un diagrama UML del elemento Baliza con sus operaciones.

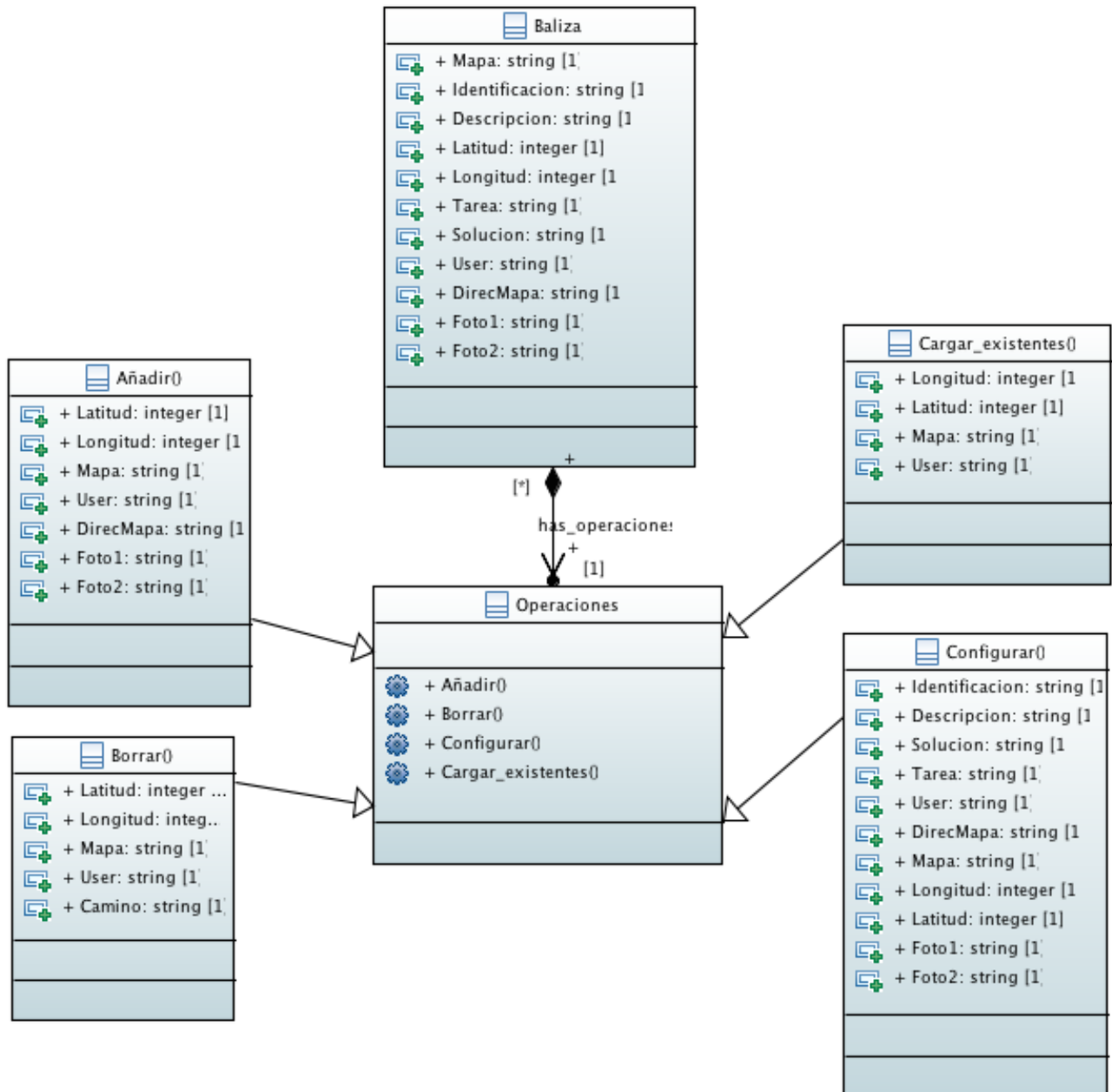


Figura 3.1 Diagrama UML del elemento baliza

En cambio, el elemento Recorrido se caracteriza de manera individual según el mapa y usuario al que corresponda, el nombre del recorrido, la distancia aproximada, el tiempo de realización, y por supuesto las balizas que lo constituyen. Además, este elemento dispone de ocho operaciones: generar, borrar, modificar añadiendo balizas, modificar eliminando balizas, cargar existentes, visualizar información de las balizas, establecer tiempo e imprimir. En la Figura 3.2 se puede observar a modo de resumen un diagrama UML del elemento Recorrido con las distintas

operaciones.

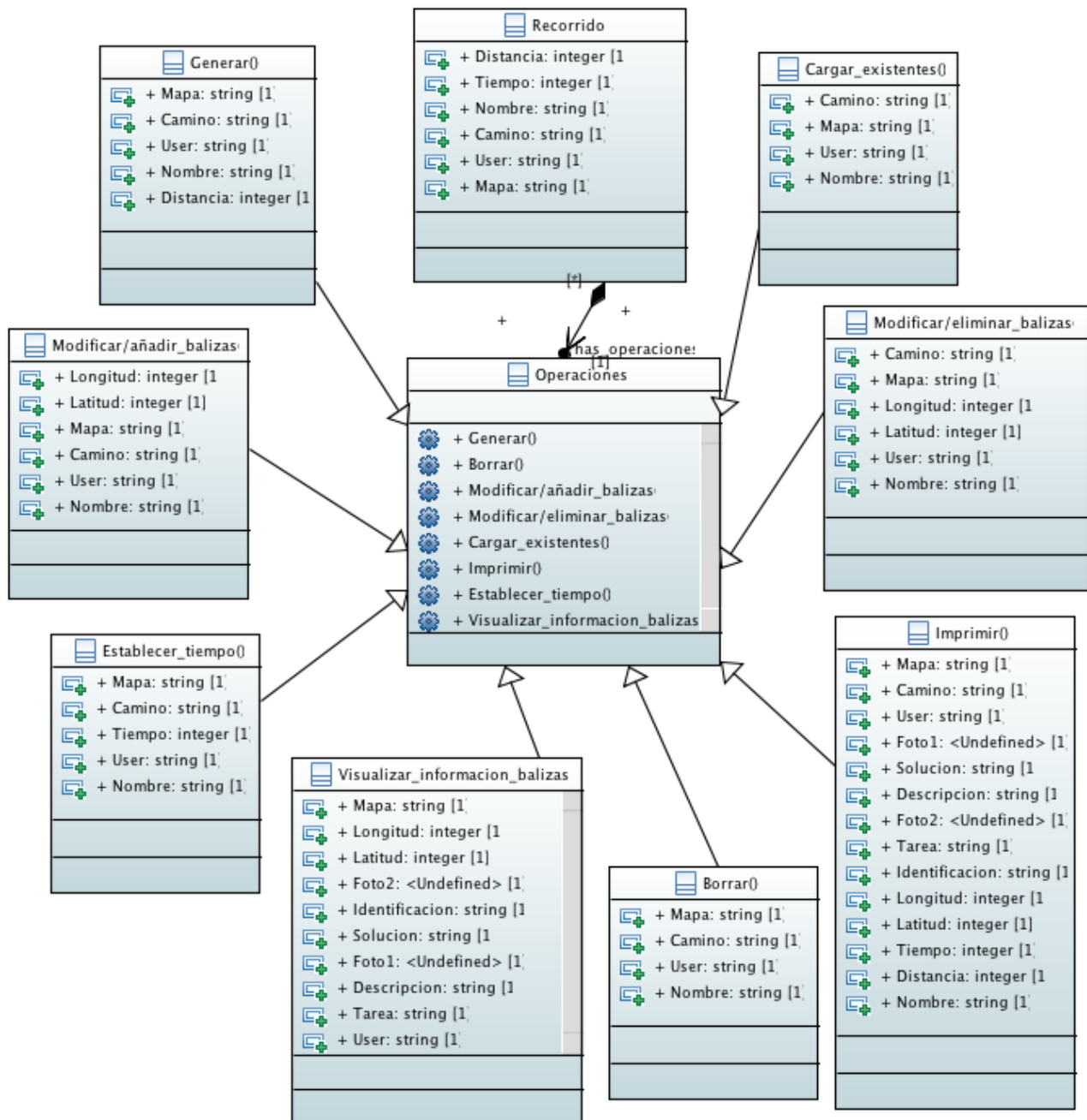


Figura 3.2 Diagrama UML del elemento recorrido

Sin embargo, es preciso destacar que tanto las balizas como los recorridos generados corresponden siempre al usuario y al mapa sobre el que los crea. De esta manera el usuario realmente genera mapas modificados sobre los cuales puede realizar las distintas operaciones con ambos elementos. Y la relación entre ambos elementos, como ya se ha comentado, es que un recorrido es un conjunto ordenado de balizas que se recogerán en la variable “camino”.

Antes de abordar el funcionamiento de la aplicación es necesario explicar la transición entre las distintas pantallas. La aplicación sólo consta de 5 pantallas independientes, para evitar un alto consumo de recursos, éstas son:

- **Inicio:** es la pantalla en la que se permite el registro de usuarios y el inicio de sesión.
- **Principal:** es la pantalla que abarca casi toda la actividad de la aplicación.
- **Configuración de baliza:** se accede a esta pantalla para realizar la configuración de las balizas.
- **Ayuda:** en esta pantalla se muestran preguntas y respuestas para resolver posibles dudas del usuario.
- **No conexión:** esta pantalla es muy importante debido a que la aplicación sin conexión a Internet no puede funcionar. Esta página será mostrada para indicar al usuario que no dispone de conexión y por tanto no puede utilizar la aplicación.

Es necesario resaltar que con pantalla nos referimos a las distintas secciones estáticas que componen la aplicación. Es decir, estas secciones no ven modificada su apariencia a lo largo de la ejecución de la aplicación a excepción de una de ellas, la principal, sobre la que se producirán transiciones entre los distintos menús que la componen (Selección de mapa, Balizas y Recorridos). En la Figura 3.3 se ilustra la interacción entre estas pantallas.

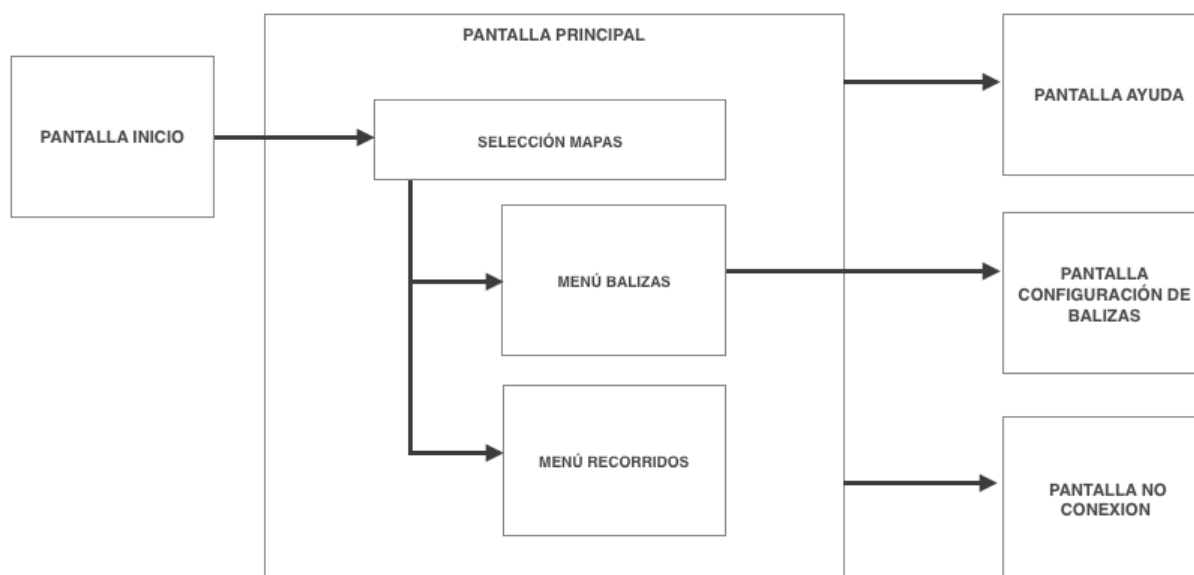


Figura 3.3 Interacción entre las pantallas de la aplicación

3.2 Funcionamiento de la aplicación

Para una mayor comprensión, en este apartado se van a explicar detalladamente las pantallas, junto con las distintas acciones que pueden llevarse a cabo sobre las mismas

3.2.1 Pantalla Inicio

Antes de pasar a explicar las posibles acciones sobre esta página, vamos a ilustrar en la Figura 3.4 un diagrama de navegación de la misma que nos ayude a comprender más fácilmente la explicación posterior.

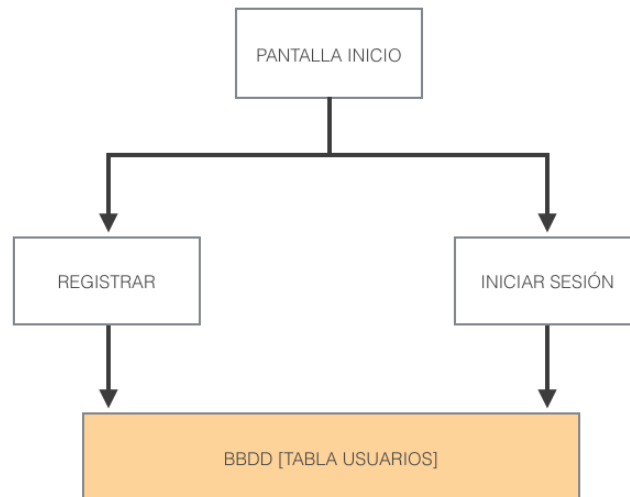


Figura 3.4 Diagrama de navegación de la pantalla Inicio

Por tanto, cuando el usuario inicia la aplicación se encuentra con la pantalla de inicio, en la que el usuario tiene dos opciones Iniciar Sesión o Registrarse. La opción Registrarse, será la primera que deberá realizar. Esta opción permite al usuario introducir a modo de formulario con etiquetas de tipo input un correo y una contraseña que deberá repetir. Para ello se ha generado una función *JavaScript* que permite comparar ambas contraseñas y a través de *AJAX* y un archivo generado en PHP se comprueba la no existencia de otro usuario con ese nombre en la base de datos. Si finalmente no existe ningún usuario con ese correo y ambas contraseñas son iguales se realiza el registro del mismo en la base de datos.

Para evitar que la contraseña se almacene sin ningún tipo de seguridad, se realiza un paso previo al registro, el encriptado de la contraseña. Para ello, se han descartado las funciones de PHP hash *md5()* y *sha1()* que no se recomiendan para la codificación de contraseñas (por la rapidez en la extracción por fuerza bruta), así que se ha empleado una función llamada *crypt()* que por defecto utiliza el algoritmo *md5*, pero por motivos de seguridad se ha utilizado *Blowfish*.

Tras el registro, se pedirá nuevamente al usuario a través de un nuevo formulario que inicie sesión con su nombre y contraseña. La primera vez que se inicia sesión se generará la base de datos interna en el dispositivo tablet para poder almacenar la información necesaria. En el Anexo A se explica detalladamente la información y estructura de la Base de Datos interna.

3.2.2 Pantalla Principal

Una vez realizado correctamente el inicio de sesión, se realiza un cambio de pantalla gracias a la función jQuery `$.mobile.changePage()` a la pantalla Principal. Nuevamente, se muestra en la Figura 3.5 un pequeño diagrama de navegación de las distintas funciones disponibles en ella.

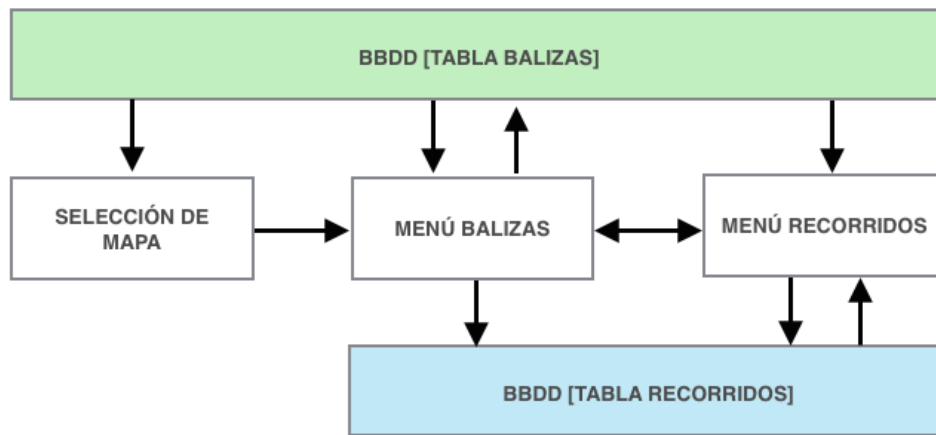


Figura 3.5 Diagrama de navegación de la pantalla principal

Una vez en esta pantalla el usuario dispondrá de 2 opciones principales a elegir: Elegir mapas modificados y Buscar por dirección, además en la esquina superior izquierda se han dispuesto dos botones: Cerrar sesión y Ayuda, tal y como puede apreciarse en la Figura 3.6.

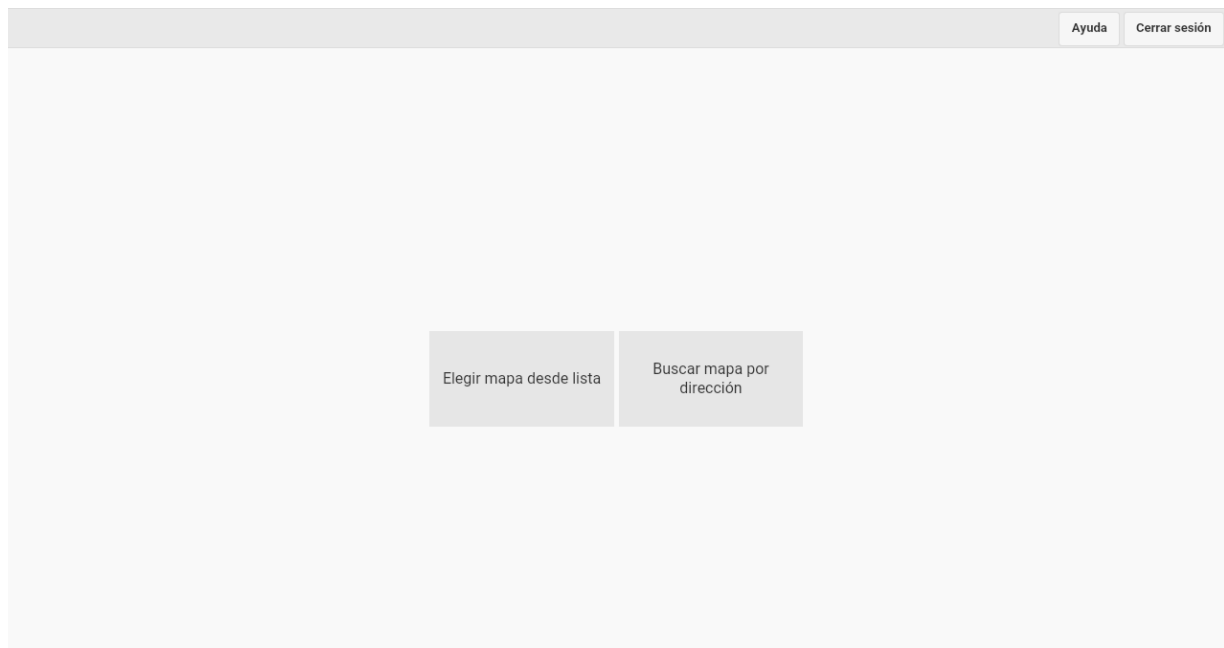


Figura 3.6 Apariencia inicial de la página principal

La opción “Buscar por dirección” permite buscar el mapa utilizando el motor de búsqueda de Google Maps. Y la opción “Elegir mapas modificados” permite seleccionar entre todos los mapas que el usuario haya generado y guardado previamente que serán solicitados mediante *AJAX* y secuencias *SQL* al servidor.

Independientemente de que se seleccione un mapa por una opción u otra, se mostrará a pantalla completa un mapa de Google estilo Satélite con centro en las coordenadas de la dirección elegida. Además, aparece un menú en la parte inferior de la pantalla con opciones como Menú Balizas, Menú Recorridos, Cambiar estilo Mapa, Guardar y salir. También se añade un botón en la esquina superior derecha “Modo parque” cuyo funcionamiento se explica a lo largo de la memoria.

Sobre esta página se irán mostrando y ocultando el resto de elementos de la aplicación ya que serán generados de forma dinámica a través de funciones *JavaScript*. En la Figura 3.7 se ilustra el formato general de esta pantalla. Y a continuación se irán explicando los distintos menús y las operaciones disponibles en cada uno.



Figura 3.7 Formato de la pantalla principal

3.2.2.1 Menú Balizas

Este menú, véase Figura 3.8, abarca todas las operaciones que corresponden al elemento Baliza nombradas anteriormente. Es decir, cualquier interacción que se desee realizar sobre una baliza deberá llevarse a cabo sobre esta sección de la aplicación.

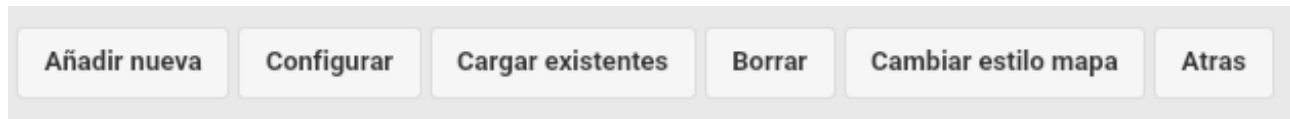


Figura 3.8 Apariencia del Menú Balizas

Este menú refleja las distintas opciones del elemento baliza, que serán explicadas a continuación. Sin embargo, a modo de resumen se dispone la Figura 3.9 donde se muestra un diagrama de navegación de este menú.

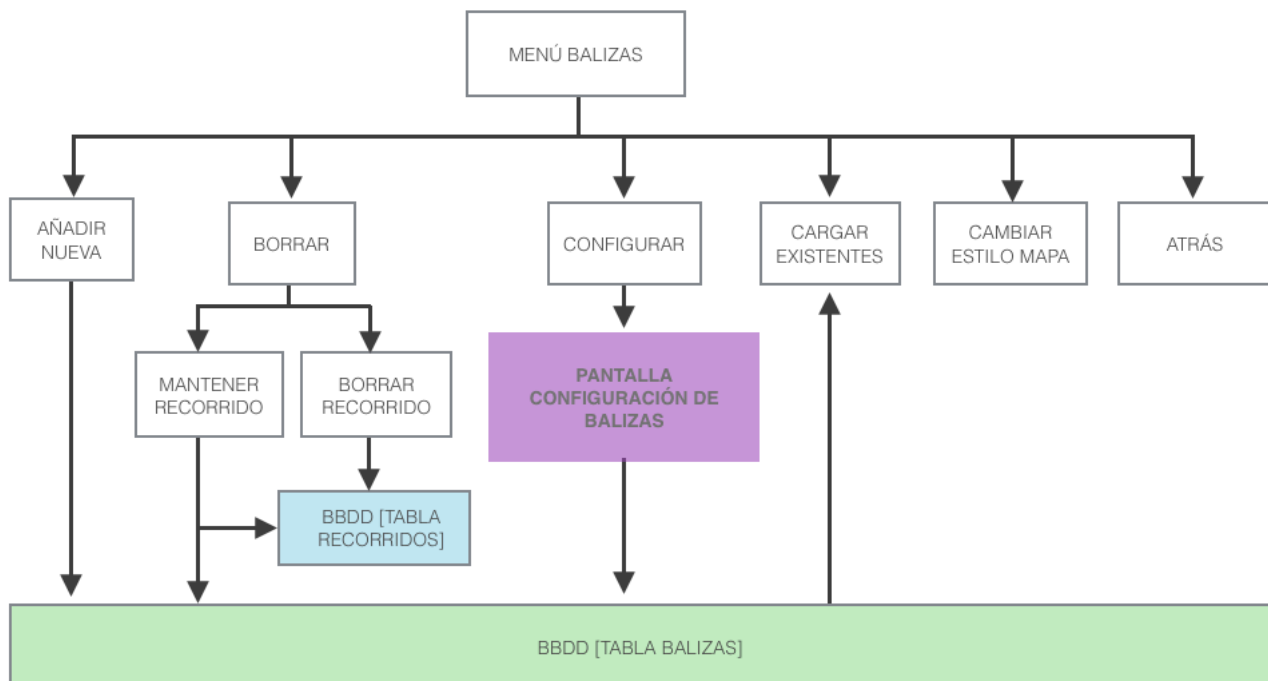


Figura 3.9 Diagrama navegación del Menú Balizas

- **Añadir Nueva:**

Al pulsar este botón generamos un *Listener* (código que se ejecuta cuando ocurre un evento determinado) asociado al mapa de Google mediante la función `google.maps.event.addListener()`. Con él lo que se pretende es que cada vez que el usuario pulse sobre el mapa se muestre un *marker* (identificador de localización), es decir, una baliza. Para ello, el *Listener* detecta la posición del click donde el usuario posiciona su dedo y la traduce a coordenadas. Con ellas genera un elemento *LatLng* (elemento propio de Google para identificar un par de coordenadas) mediante la función `google.maps.LatLng()`. Una vez identificadas las coordenadas se posiciona la baliza en esa posición con un icono verde y con texto “-“, tal y como se muestra en la Figura 3.10. Este icono se solicita mediante un enlace a esa imagen.

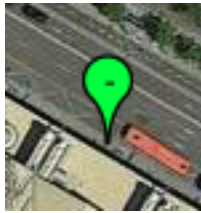


Figura 3.10 Apariencia de una baliza posicionada sobre el mapa

El usuario puede posicionar tantas balizas como desee y siempre mientras no pulse ningún otro botón, pues en ese momento será eliminado el *Listener* y el mapa dejará de detectar la posición del dedo. Cada vez que el usuario posiciona una baliza, ésta se almacena en la base de datos del servidor y se genera un nuevo *Listener*, esta vez asociado a las balizas, que detecte cuando se hace click en alguna de ellas. Este nuevo *Listener* gestiona las distintas acciones programadas sobre una misma baliza (configurar, borrar, etc), y para determinar que acción es ejecutada en cada momento, se han creado unas variables booleanas (“del”, “info”, “numer”, etc) que hemos denominado “variables de marker”, y nos permiten elegir entre los distintos códigos a ejecutar.

- **Configurar:**

Esta opción, supone en primer lugar, el cambio de la variable “info” a “true” y el resto de variables de marker a “false”, permitiendo que se ejecute únicamente el código generado para la configuración de la baliza. En segundo lugar, una transición a la página de configuración. Y en tercer lugar el almacenamiento de la posición de la baliza seleccionada en una variable.

- **Cargar existentes:**

Cuando se pulsa este botón, se solicitan todas las balizas que existen en el mapa seleccionado de ese usuario concreto. Para ello, mediante una función *AJAX* se envía el nombre del mapa y el usuario al servidor, éste interpreta la información recibida con un archivo *PHP* generado para ello, en él se han programado secuencias *SQL* que permiten solicitar a la base de datos dicha información. Una vez se recibe la información sobre las balizas, se lee y genera una variable *LatLng* con la posición de cada una de ellas. Y mediante el posicionamiento de nuevos *Markers* se muestran con el mismo formato y con el mismo proceso que las balizas nuevas. Sin embargo, en este proceso no se almacenan las balizas nuevamente en la base de datos. El proceso queda ilustrado en la Figura 3.11.

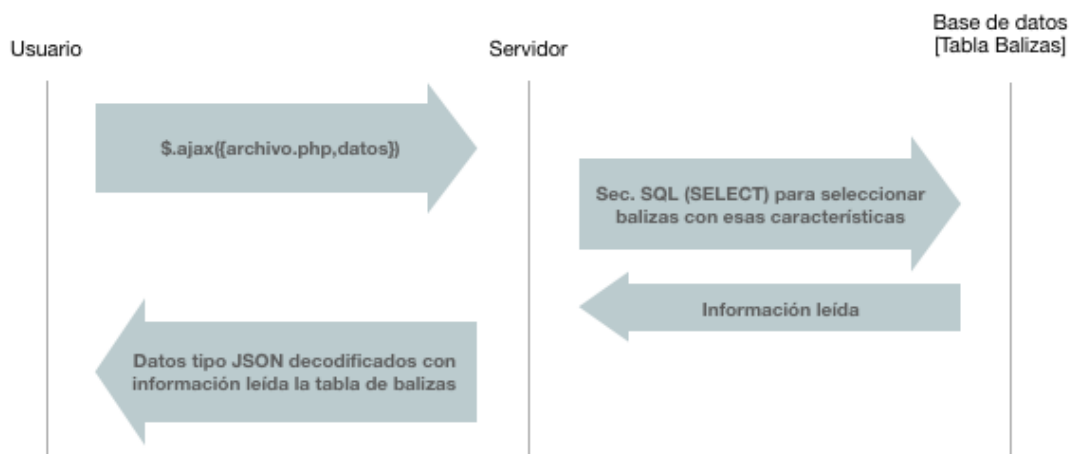


Figura 3.11 Proceso Cargar Balizas

- **Borrar:**

Cuando se acciona esta opción, se cambia el valor de la variable “del” a “true” para seleccionar el código de borrado en el *Listener* asociado a los *Markers* y el resto de variables de marker las cambiamos a “false”. Por tanto, al seleccionarse la baliza que se desea borrar se almacena en una variable la posición de la misma. Cabe destacar, que esta variable es un objeto y para su utilización en una secuencia *SQL* debemos previamente convertir a *String* (cadena de caracteres). Una vez seleccionada, se permiten dos opciones de borrado, “Borrar eliminando también los recorridos” y “Borrar la baliza manteniendo los recorridos”.

La primera de las opciones es bastante clara, puesto que borrará tanto la baliza seleccionada como todos los recorridos del usuario siempre que estén sobre el mapa seleccionado y contengan esa baliza. En este proceso se envían con *AJAX* las coordenadas de la baliza, el nombre del mapa y el usuario. El servidor interpretará la información a través de un archivo *PHP* generado previamente, y mediante secuencias *SQL* contenidas en él, se solicita el borrado en la base de datos de la baliza seleccionada y de aquellos recorridos que contengan el usuario, el mapa seleccionado y las coordenadas de la misma en su camino.

La segunda opción supone que la aplicación eliminará la baliza de la base de datos, pero no eliminará los recorridos que la contengan. En su lugar, se eliminarán las coordenadas de la baliza del recorrido al que pertenecen. Nuevamente comprobando previamente que dicho recorrido y baliza correspondan al usuario y mapa seleccionado.

Como la operación “Cambiar estilo mapa” se muestra permanentemente sobre la barra de menús he decidido explicarlo de forma común en el apartado 3.2.2.3.

3.2.2.2 Menú Recorridos

Este menú recoge todas las acciones que corresponden al elemento Recorrido que han sido

nombradas anteriormente. Por tanto, cualquier interacción que desee realizarse sobre un recorrido deberá llevarse a cabo sobre este apartado de la aplicación.

Figura 3.12 Apariencia del Menú Recorridos

Este menú, véase Figura 3.12, muestra las distintas opciones del elemento Recorrido, que van a ser explicadas en las siguientes líneas. Sin embargo, a modo de resumen se incluye la Figura 3.13 donde se muestra un diagrama de navegación de este menú.

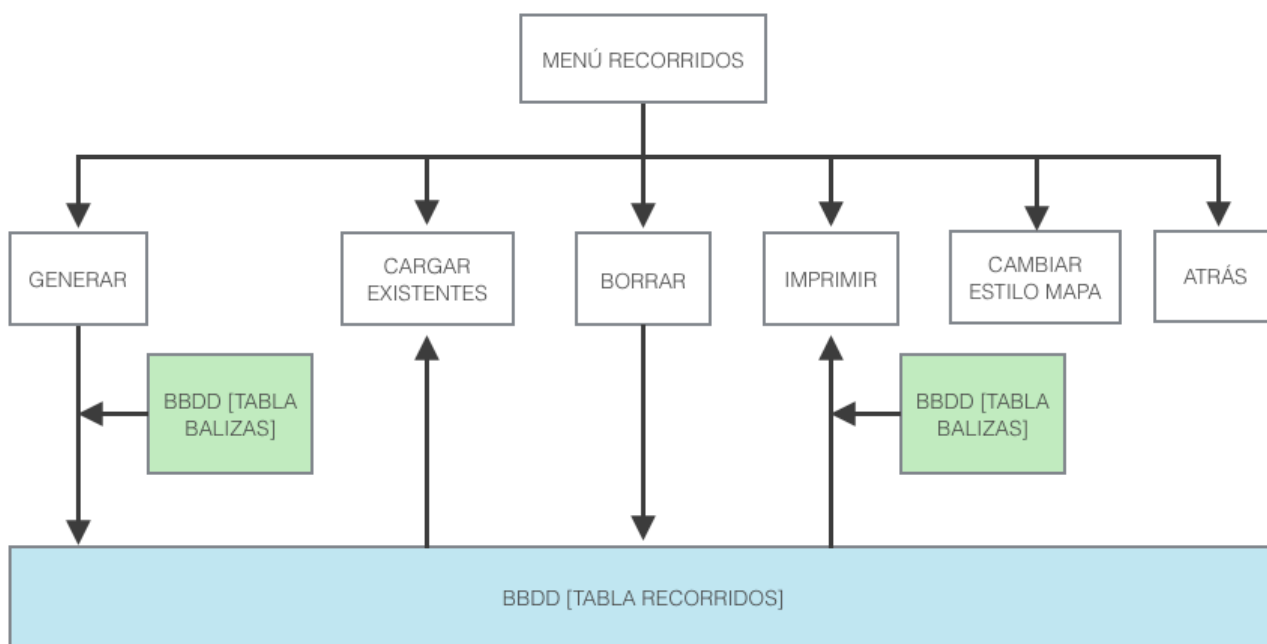


Figura 3.13 Diagrama navegación del Menú Recorridos

- **Generar:**

Al seleccionar esta opción, inicialmente se activa la variable “numer” y se desactivan el resto de variables de marker. De este modo, se permite la numeración de las balizas. Por lo tanto, cuando el usuario vaya pulsando cada baliza que desee incorporar al recorrido, las coordenadas de la misma se almacenarán en un vector que conformarán todo el camino del recorrido. Una vez asociado un nombre al recorrido, se envía esta información con *AJAX* al servidor que la interpretará gracias a un archivo generado en PHP y a través de secuencias SQL se registrará en la base de datos el nuevo recorrido, con características como el nombre del mismo, el camino de balizas, el nombre del mapa, la distancia y el usuario.

- Cargar existentes:

Cuando esta opción es seleccionada, ocurre algo similar a lo ocurrido con las balizas. Es decir, se solicita la lectura mediante *AJAX* y un archivo programado en *PHP* de todos aquellos recorridos que hayan sido generados por el usuario para ese mapa concreto. Una vez recibida la

información, deben leerse todos los campos. Se generará dinámicamente una lista (etiquetas `` y ``) con los nombres de los recorridos existentes. De esta forma, un usuario podrá elegir el recorrido que desea visualizar pulsando sobre el nombre del mismo.

No obstante, para ello es necesario prestar especial atención al campo “Camino” (campo que incluye las coordenadas de las balizas que lo forman) en la lectura de la base de datos. Este campo, una vez leído, debe ser separado por parejas de latitud y longitud.

Una vez tenemos las posiciones de las balizas, debemos separarlas nuevamente en dos, latitud por un lado y longitud por otro. Esto se debe hacer así porque, para poder generar los elementos *LatLng*, necesarios para colocar un nuevo *Marker* en su posición, se requiere introducir los dos parámetros por separado.

De esta manera, solo nos queda llevar a cabo este proceso de forma dinámica hasta posicionar en su lugar todas las balizas que forman el recorrido elegido. En la Figura 3.14 se muestra un pequeño proceso ilustrativo.

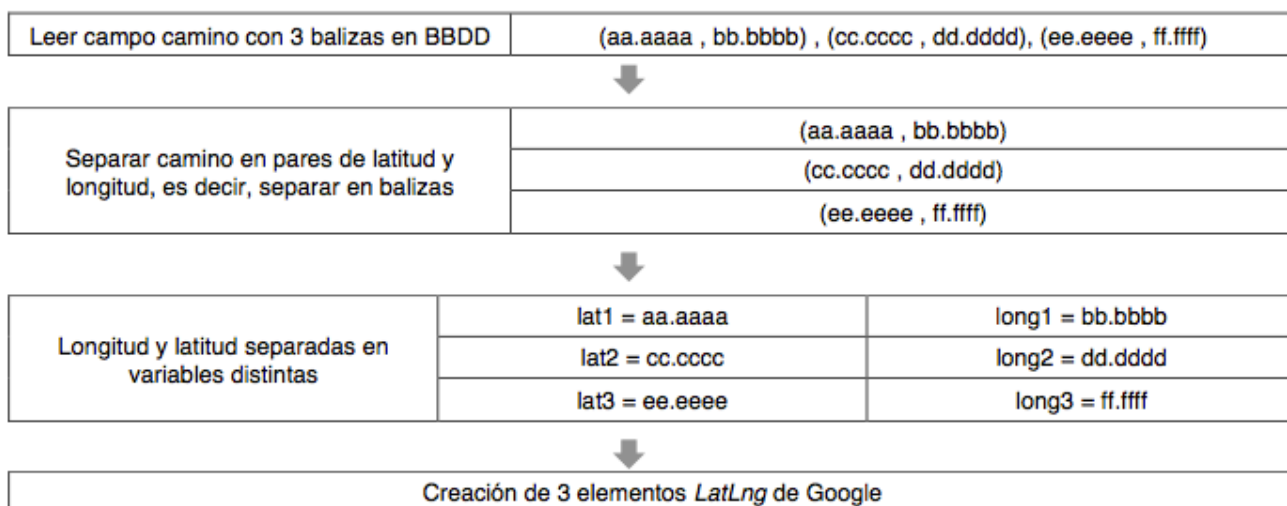


Figura 3.14 Ilustración proceso división coordenadas

Además a modo de ayuda visual, sobre el mapa se incluye un elemento *Polyline*, propio de Google, con la función `google.maps.Polyline()`. Este elemento no es más que una línea que une dos puntos. Para su visualización en el mapa, es necesario guardar en dos variables la posición inicial y final de la línea, siempre en pares de dos balizas.

- **Borrar:**

Al seleccionar esta opción, se activa en el código la variable de marker llamada “dele”, ésta está relacionada con un *Listener* asociado a cada objeto *polyline*. De esta manera, al seleccionar un objeto *polyline*, se ejecuta una función que depende de él y mediante la función `getPath()` asocia el camino que dibuja el objeto *polyline* con una variable. Una vez hecho esto, se convierte a *String* cada segmento del recorrido y se almacena en un array que posteriormente será enviado mediante *AJAX* al servidor para que gracias a un nuevo archivo *PHP* generado,

elimine de la base de datos el recorrido seleccionado. Una vez eliminado, se borra el recorrido tanto del mapa con la función *setMap(null)* como de la lista de recorridos con la función *remove()*.

- **Imprimir:**

Al seleccionar esta opción se pueden generar 2 tipos de documentos: para profesor y para alumno (con 2 versiones). En este proceso pueden distinguirse varios pasos:

El primero, es generar a partir de la información del recorrido (el orden y la posición de las balizas) un mapa estático con el estilo de mapa que estuviéramos utilizando en ese momento. Este tipo de mapas, al ser imágenes permiten su posterior impresión por lo que es imprescindible su generación.

Simultáneamente, se ejecuta una sección de código programada para generar una tabla de información y una vez hecha, una sección de código desarrollada permite rellenar la tabla con la información correspondiente de cada baliza. Este proceso puede demorarse unos segundos puesto que debe realizarse completamente de forma dinámica. Esto es así para evitar almacenar información de las balizas en variables y consumir mayor número de recursos.

En esta sección, se han programado dos botones que permiten cambiar el formato del documento entre tipo profesor, con toda la información rellenada y el tipo alumno, con los campos “Solución” vaciados y con un nuevo mapa estático sin balizas. Ambos tipos de documentos se disponen en 2 versiones: un informe completo con toda la información y uno compacto con información sobre la descripción, la tarea y la solución que puede o no incluir fotos.

Para evitar el descuadre de la tabla cuando las balizas no incluyan fotografías, se ha generado una imagen que será añadida directamente a la baliza cuando ésta sea generada la primera vez. Además cada vez que se deba generar una nueva tabla sobre esta pantalla, la anterior será eliminada para evitar que la información se solape de un informe a otro.

Una vez generado el informe deseado, al final del mismo se muestra en todos los casos un botón que nos permite el envío por correo del archivo .pdf. Una vez lo presionemos, se lleva a cabo la generación del documento .pdf a través de funciones del plugin *html2pdf*. Estas funciones están programadas en un archivo *PHP* almacenado en el servidor al cual habrá que pasarle la información a imprimir mediante *AJAX*.

Esta información incluye, el mapa estático y la tabla de información, así como el correo al que enviaremos el archivo que se leerá de la base de datos interna del teléfono. Cabe destacar que la dirección de envío se puede cambiar y en ese caso, no será leída de la base de datos si no que el usuario rellenará un campo “email” que será leído y transmitido al servidor.

Esta información es recibida e interpretada por un archivo *PHP* anteriormente nombrado. En este archivo, se establecen también los márgenes de las páginas. Una vez

generado el archivo .pdf, se emplean funciones del plugin *swiftmailer* que permiten el envío del correo electrónico a la dirección especificada a través del servidor SMTP (Simple Mail Transfer Protocol) configurado.

Hasta el momento, se han explicado todas las operaciones del menú Recorridos, pero tal y como se ha ilustrado previamente en la Figura 3.2 existen cuatro operaciones de recorrido adicionales: establecer un tiempo de duración, visualizar información sobre las balizas, y modificar el recorrido eliminando balizas o añadiendo balizas,. Estas operaciones se encuentran sobre la pantalla principal, y a modo de ilustración se añade la Figura 3.15 donde se puede observar su localización:

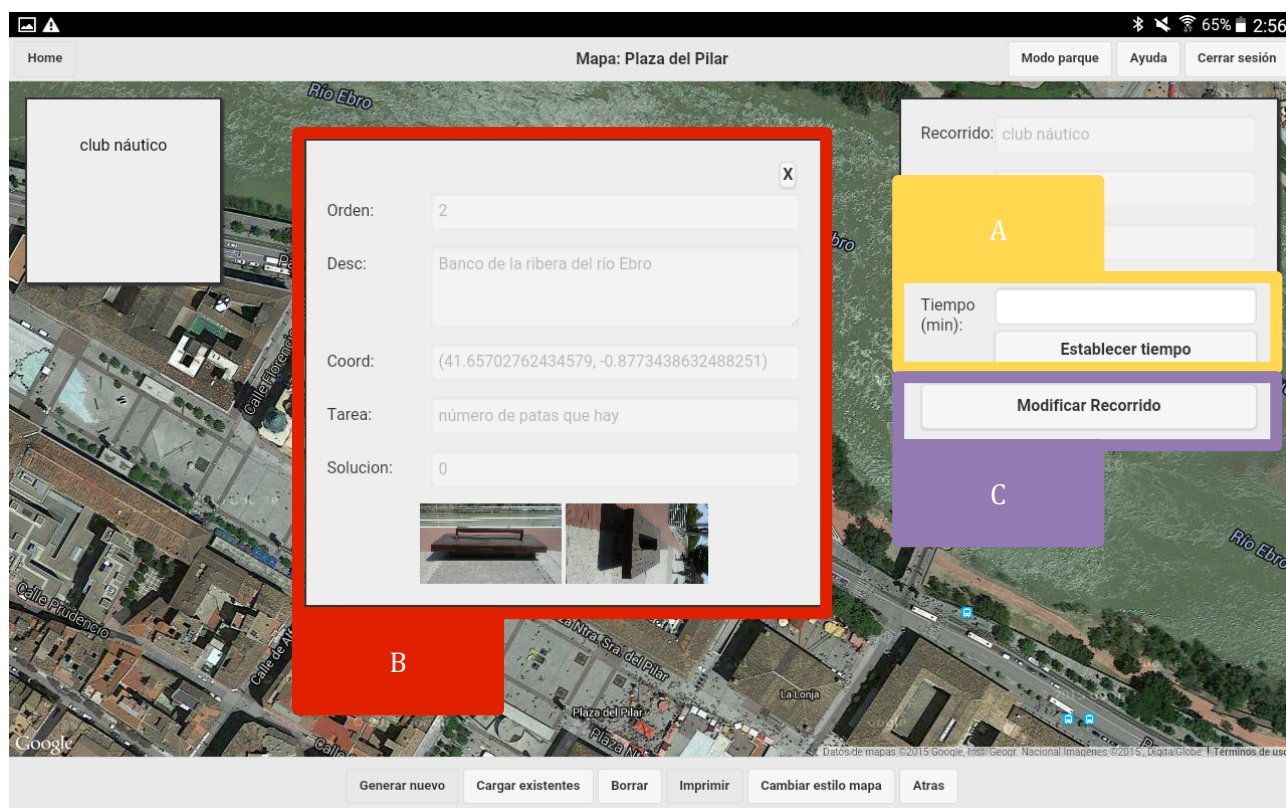


Figura 3.15 Funciones adicionales de los recorridos

- A. Esta opción permite dotar al recorrido de un tiempo aproximado de duración. De esta manera, el usuario debe rellenar formulario con una etiqueta tipo *input* que permita modificar sobre la base de datos el campo “tiempo”. Esto se realizará enviando la información a través de *AJAX* al servidor que gracias a un archivo *PHP* programado con secuencias *SQL* que permiten modificar la base de datos.
- B. Para poder visualizar este pequeño cuadro resumen con información se debe pulsar sobre cada baliza. Para poder hacer esto, se ha programado una nueva variable de marker, “ver” y nos permite abrir un cuadro de información cuando seleccionamos una baliza.
- C. En segundo lugar, existe la posibilidad de modificar el recorrido una vez generado o cargado desde la base de datos. Esto es, añadir o eliminar balizas de un determinado

recorrido mostrado sobre el mapa. De nuevo, distinguimos dos opciones:

- Añadir balizas al recorrido:

Al pulsar esta opción, se acciona el mismo código que al seleccionar “Cargar Existentes” del menú Recorridos con la salvedad de que las nuevas balizas, es decir, aquellas que no pertenecen en ese momento al recorrido, se cargarán con el icono en verde y la etiqueta “-“, mientras que las que pertenecen al recorrido no se verán modificadas. Cada vez que el usuario selecciona una baliza nueva, la posición de éstas se irá agregando a un array que permitirá la modificación correcta del recorrido. Una vez se ha terminado de seleccionar las balizas, se solicitará al usuario que introduzca el orden deseado para cada baliza asociando cada nueva etiqueta con una posición. Es necesario aclarar que una baliza nueva sustituirá la posición de la baliza antigua, y está adoptará la siguiente posición. Para una mayor comprensión se puede visualizar el proceso en la Figura 3.16.

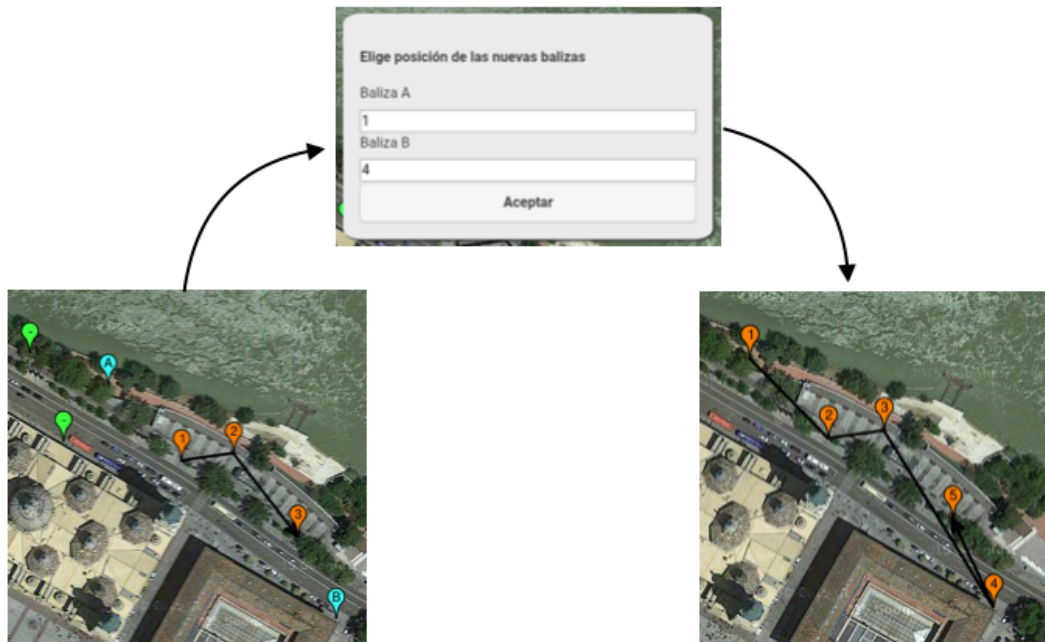


Figura 3.16 Recorrido modificado al añadir 2 balizas

Una vez se ha modificado el recorrido, se actualiza la base de datos con el nuevo recorrido que será enviado al servidor a través de *AJAX* y leído e interpretado por un archivo *PHP* programado con secuencias *SQL* para ello. Y finalmente se actualiza el recorrido sobre el mapa.

- Eliminar balizas del recorrido:

En esta opción se permite que el usuario seleccione las balizas del recorrido

que desea eliminar. Para ello, nuevamente se activa el *Listener* de los markers y cuando el usuario selecciona las balizas a eliminar, las coordenadas de éstas se almacenarán en un vector. Y una vez finalizada la modificación del recorrido, se procede a eliminar sobre el trayecto antiguo todas las coordenadas almacenadas en el vector de borrado. De esta manera, se modifica el trayecto y se almacena en la base de datos nuevamente a través de *AJAX* y gracias a un archivo *PHP* almacenado en el servidor programado para ello. Además de forma dinámica, se actualiza el recorrido de forma visual, es decir, una vez modificado, el usuario ve sobre el mapa únicamente las balizas que forman el nuevo recorrido actualizado.

Cabe destacar que en este menú no están activas ninguna de las variables de marker utilizadas en las operaciones del menú balizas y por tanto para realizar alguna función relacionada con ellas: añadir, borrar o modificar su información es necesario llevarla a cabo desde ese otro menú porque es necesaria la activación de esas variables de marker que permitirán llevar a cabo estas operaciones.

Nuevamente comentar que como la operación “Cambiar estilo mapa” se muestra permanentemente sobre la barra de menús he decidido explicarlo de forma común en el apartado 3.2.2.3.

3.2.2.3 Cambiar estilo Mapa

Esta opción permite cambiar el estilo del mapa de Google entre sus cuatro versiones:

- MapTypeId.ROADMAP: es el tipo de mapa por defecto que muestra Google.
- MapTypeId.SATELLITE: muestra un mapa estilo satélite.
- MapTypeId.HYBRID: este tipo es una mezcla de los dos primeros tipos.
- MapTypeId.TERRAIN: muestra un mapa físico con información del terreno.

Estos estilos, se van cambiando en bucle. Para elegir el tipo de mapa deseado, solo es necesario ir pulsando este botón.

3.2.2.4 Guardar y salir

En esta opción, lo que se realiza es el guardado de toda la información relacionada tanto con las balizas como con los recorridos en la base de datos externa. De no pulsarla, se realizará el borrado de las balizas y recorridos generados desde el ultimo guardado. Para poder realizar esto, se requiere la lectura del índice de la última baliza y el último recorrido y almacenarlos en la base de datos interna cada vez que el usuario inicie sesión. De esta manera, si la aplicación no guarda las balizas, se lleva a cabo el borrado de todas las balizas y recorridos nuevos (con índice mayor al guardado) que correspondan con ese usuario.

3.2.2.5 Modo parque y botón Home

Hasta el momento, se han descrito varias necesidades que un profesor podría necesitar a la hora de crear un mapa de orientación para sus alumnos. No obstante, tanto para la colocación de las balizas como para la realización de las fotografías que las caractericen, el profesor deberá desplazarse indiscutiblemente al parque. Con el objetivo de facilitar esta tarea al profesor un poco más, se ha añadido finalmente una nueva función a la aplicación, “Modo parque”.

Ésta consiste únicamente en mostrar sobre el mapa la posición actual del profesor mediante GPS y permitirá que éste pueda visualizar de manera sencilla si la baliza queda correctamente colocada o si las fotos que realiza se corresponden con la baliza adecuada.

Para poder llevar a cabo esta nueva función, es necesario acceder a la función GPS del dispositivo móvil, y para ello se ha utilizado el plugin de localización que proporciona Phonegap. Éste a través de la función *navigator.geolocation.watchPosition()* nos proporciona las coordenadas de la posición del dispositivo. Una vez obtenidas es necesario separar en latitud y longitud para finalmente crear un nuevo objeto *LatLng* que, como ya sabemos, es necesario para poder generar un *marker* en una posición determinada.

Esta función de localización puede ser configurada para repetir en un período de tiempo la búsqueda, en nuestro caso esta repetición se realiza cada 30s. En cada repetición no se añadirá un nuevo *marker* si no que el actual únicamente verá modificada su posición. Por tanto siempre que tengamos el modo parque activo podremos visualizar un *marker*, que en este caso será un punto azul sobre el mapa que se irá desplazando a medida que el dispositivo se desplace.

Por último, en esta pantalla existe un botón “Home” que permite hacer una transición a la página Inicio.

3.2.3 Pantalla Configuración de Baliza

En esta pantalla se presenta un formulario con entradas input para que el usuario introduzca la información asociada a esa baliza. Esta información será recogida y enviada al servidor a través de *AJAX*. Un archivo almacenado en el servidor, generado en *PHP* con secuencias *SQL* recogerá la información y se actualizará la información de la baliza contenida en la base de datos. Nuevamente, en el archivo *PHP* se han añadido secuencias *SQL* para comprobar que la baliza configurada corresponde al usuario y mapa seleccionado.

Además se dispone de dos botones para asociar dos fotografías a la baliza, que podrán ser realizadas mediante la cámara o elegidas de la galería del dispositivo. En este punto es preciso destacar que las imágenes no son almacenadas en la base de datos directamente, para evitar el exceso de información en la misma. Sin embargo, lo que se realiza es una transformación de la imagen a *base64*, y ésta se transfiere al servidor donde será guardada para su uso, por tanto en la base de datos se almacena únicamente el nombre de la imagen. Esto se realiza así para evitar también que el usuario elimine la fotografía de su dispositivo y ésta deje de visualizarse. Este

proceso se ilustra para una mayor comprensión en la Figura 3.17.

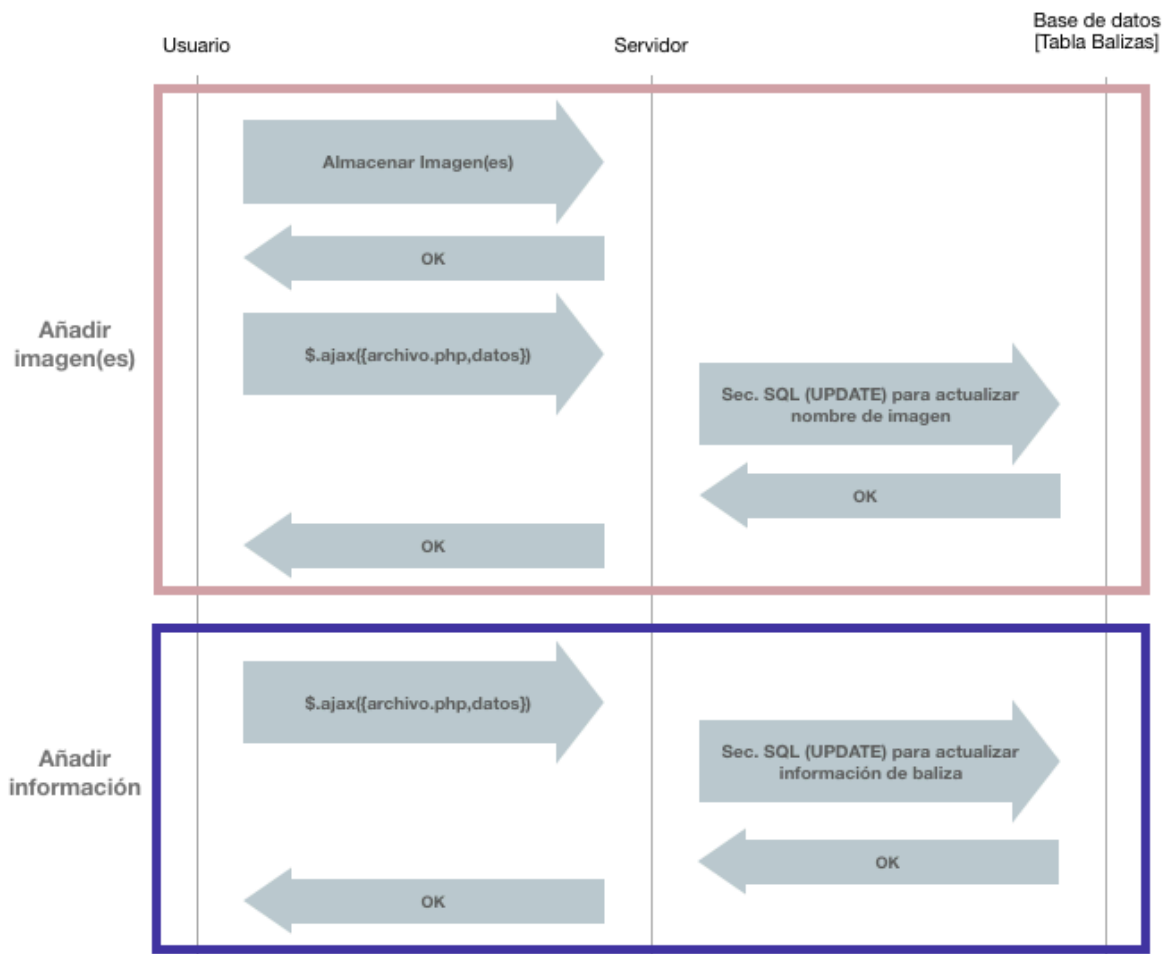


Figura 3.17 Proceso de configuración de las balizas

3.2.4 Pantallas Ayuda y no Conexión

A modo de pequeño tutorial se ha programado una pantalla con preguntas y respuestas que permiten al usuario conocer como navegar por la aplicación y llevar a cabo las distintas operaciones. Y la última pantalla desarrollada es una pantalla que se muestra siempre que la aplicación no disponga de internet. Para ello, se controla constantemente el estado de la conexión a través de funciones propias de Phonegap. Esto sirve de ayuda para notificar al usuario que no puede utilizar de forma correcta la aplicación pues es imprescindible el uso de internet en todo momento. Mientras que el usuario no disponga de internet visualizará únicamente esa pantalla al ejecutar la aplicación.

Capítulo 4

Descripción del servidor

En este capítulo se describe el servidor externo necesario para el correcto funcionamiento de la aplicación. En primer lugar se presenta un planteamiento general del mismo y posteriormente se explica de manera breve la base de datos alojada en el mismo.

4.1 Planteamiento general

Una de las partes importantes del sistema, es la existencia de un servidor externo. Era imprescindible que éste fuese capaz de comunicarse directamente con la aplicación de la Tablet a través de Internet y gestionar la base de datos. Además debía ser capaz de actuar como servidor SMTP capaz de enviar correos electrónicos.

Para ello se decidió utilizar llamadas *AJAX* a ficheros *PHP* programados previamente y almacenados en el servidor para la comunicación y como base de datos se ha utilizado una base de datos *MySQL*.

Se han requerido múltiples archivos *PHP* que permitiesen abarcar las distintas funciones que debía cumplir la aplicación. En estos archivos *PHP* están programadas las secuencias *SQL* para comunicarse con la base de datos y para evitar ataques de inyección *SQL* se han llevado a cabo medidas como el escape de caracteres.

4.2 Funcionamiento del servidor

Se perseguía una interacción con la aplicación y la base de datos totalmente óptima. Para la comunicación entre cliente y servidor se han utilizado llamadas *AJAX* con un tipo de datos llamado *JSON*. Este tipo de datos permite asignar diferentes variables a diferentes nombres proporcionando una estructura parecida al lenguaje *XML* pero con la información un poco más simplificada .

Al trabajar con Phonegap, la programación de la aplicación y la comunicación con el servidor es similar a la propia de una página web ya que la programación se hace bajo HTML, JavaScript y CSS. Por ello, para la comunicación se ha empleado *AJAX*, que hace posible realizar peticiones al servidor y obtener respuesta de este en segundo plano y usar esos datos para, a través de JavaScript, modificar los contenidos de la aplicación creando efectos dinámicos y rápidos.

Por un lado, el servidor almacena los archivos *PHP* programados que son necesarios para el correcto funcionamiento de la aplicación y comunicación con el servidor y la base de datos. Y también es en el servidor donde se almacenan las imágenes que los usuarios asocian a las balizas.

Por otro lado, se ha configurado el servidor como servidor SMTP, es decir para permitir el correcto envío de correos electrónicos. Para ello se ha generado una cuenta de Gmail para este trabajo que será configurada en el archivo encargado del envío de correos en las opciones del plugin *SwiftMailer* como dirección de envío.

4.3 Base de datos

Una parte importante de la aplicación es la base de datos en la que se almacena toda la información generada. Para un mayor control y organización de la información, se optó por generar tres tablas. Estas tres tablas, recogen información referente a las balizas creadas, los recorridos generados y los usuarios registrados. Las tres, están ordenadas por un índice que se incrementa en uno cada vez que se añade alguna fila nueva (una baliza, un recorrido o un usuario). Entre ellas, existe una relación total. Es decir, con esta estructura se persigue que pese a tener la información separada y ordenada de esta manera, siempre exista una relación directa entre ellas. Por lo tanto, el campo “user” de las tablas “balizas” y “recorridos” será necesariamente el correo de un usuario registrado en la tabla “usuarios”. De la misma manera, en la tabla “recorridos” el campo “recorrido” estará formado por balizas existentes en la tabla “balizas”.

En el Anexo A se puede encontrar una explicación más detallada de las tablas de la base de datos.

4.3.1 Tabla Balizas

Esta tabla es la tabla que almacena información sobre todas las balizas que se posicionan sobre los distintos mapas. Dispone de 12 columnas siendo estas: id, baliza, user, descripción,

coorBaliza, tarea, solución, mapa, direcMapa, foto1, foto2 y elim.

De esta manera, podemos almacenar toda la información de cada baliza añadiendo una fila nueva a esta tabla. Todas las balizas de todos los usuarios se almacenan en ella y se distinguen de un usuario a otro por el nombre de usuario.

4.3.2 Tabla Recorridos

La información sobre los recorridos generados se almacenan en esta tabla. Esta formada por 10 columnas, que son: id, mapa, recorrido, user, nombre, distancia, tiempo, antiguo, modif, y elim.

En ellas podemos almacenar toda la información de cada recorrido añadiendo una fila nueva a esta tabla. De la misma manera que antes, todos los recorridos de todos los usuarios se almacenan en ella y se distinguirán de un usuario a otro por el nombre de usuario.

4.3.3 Tabla Usuarios

En esta tabla, se almacena la información relativa a los usuarios registrados. Esta formada por 3 columnas, id, correo y password.

El campo password mostrará la contraseña del usuario siempre de forma cifrada puesto que este cifrado se realiza previamente al registro del usuario en la base de datos.

Capítulo 5

Pruebas y evaluación

En este capítulo se describen las distintas pruebas que se han ido realizando a medida que el desarrollo de la aplicación avanzaba. Estas pruebas, se han dividido entre las realizadas durante y tras el desarrollo de la misma.

5.1 Pruebas durante y tras el desarrollo

En el proceso de desarrollo se han ido realizando diversas pruebas. Éstas se realizaban tanto en un emulador de Android desde la plataforma Eclipse, cómo en un dispositivo tablet, una Samsung Galaxy Tab 4, que se adquirió para el desarrollo del TFG.

Con estas pruebas, se buscaba en todo momento un mayor aporte de información en cuanto a aspecto visual de la interfaz como el cumplimiento de los requisitos de la aplicación. Además, durante la etapa de desarrollo de la aplicación se han ido obteniendo errores de utilización que ha sido necesario solventar para una mayor comodidad de uso para el usuario. Y tras su realización fueron necesarias varias pruebas de uso que permitiesen comprobar que la programación era correcta y su uso era dinámico y práctico.

Se ha mostrado la aplicación durante diversas fases del desarrollo a miembros de la comunidad de EFYPAF para poder recibir un feedback sobre las distintas utilidades de la aplicación. En todo momento, se perseguía la obtención de una herramienta útil y versátil que ayudara las tareas del profesor de Educación Física, y la valoración de los miembros de esta comunidad nos resultó de gran utilidad.

Por otro lado, se han realizado pruebas en diferentes dispositivos para comprobar su correcto funcionamiento. Como en todo proyecto, tras su realización son necesarias varias pruebas de uso que permitan comprobar que su programación es correcta y su uso es dinámico y práctico.

Sin embargo, el trabajo posterior a la aplicación no termina una vez esta está actualizada con las nuevas funcionalidades que se le puedan aportar, sino que es necesario ir retroalimentándola periódicamente para otorgarle una mayor usabilidad.

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se describe una pequeña opinión personal sobre el proyecto realizado así como las distintas conclusiones que pueden ser extraídas de su realización y también un pequeño análisis de las posibles líneas que podría seguir el proyecto tras su realización.

6.1 Opinión personal

La realización de este proyecto TFG ha supuesto todo un reto a nivel personal para mí por varias razones. La primera y más importante porque supone la finalización del grado. La segunda, porque pese a que tenía conocimientos básicos de programación HTML, JS y PHP, sumergirme en su conocimiento y aprender tecnologías nuevas como AJAX o JSON, a parte de ser un esfuerzo personal ha supuesto un enriquecimiento de mis competencias profesionales. Por otro lado, uno de mis objetivos para con este trabajo era desarrollar una aplicación que no se quedara meramente en un trabajo de fin de grado sino que este pudiese convertirse en una herramienta útil para, en mi caso, profesores de Educación Física y creo que cumple con los requisitos.

Para concluir, creo que este trabajo ha sido una gran experiencia personal, puesto que en su realización he disfrutado con todos los conocimientos que iba aprendiendo así como con el resultado obtenido.

6.2 Conclusiones

Este trabajo perseguía el desarrollo de una aplicación funcional en Phonegap, capaz de ser

exportada a todas las plataformas de forma rápida, y que supusiese una herramienta útil para la generación de recorridos para profesores de Educación Física.

Para su realización se han abordado conocimientos como la programación en Phonegap, HTML, JS y PHP, el diseño de la interfaz visual mediante hojas de estilos CSS, la interacción con la API de Google, el uso de las bases de datos, la transferencia de archivos a un servidor y la generación y envío de un documento con formato .pdf.

Finalmente, teniendo en cuenta el resultado obtenido, y las valoraciones recibidas por miembros de EFYPAF se puede concluir que los objetivos establecidos al inicio han sido alcanzados y por tanto podemos determinar que el proyecto de fin de grado ha sido realizado de forma exitosa.

6.3 Líneas futuras

Llevar a cabo una evaluación de los usuarios finales puede suponer la primera de las líneas futuras ya que esta información puede aportarnos algunas mejoras y nuevas funcionalidades que pueden ser añadidas a la aplicación. Ésta podrá ser utilizada por profesores de Educación Física que es el objetivo perseguido con el desarrollo de la aplicación. y una vez que el trabajo sea evaluado, la aplicación será subida a Google Play para que los usuarios puedan descargarla de forma gratuita.

Además, su desarrollo en Phonegap, nos permite su exportación a otras plataformas como por ejemplo iOS y la posibilidad de subirla a iTunes. También, se podría adaptar y exportar su utilización al campo del turismo donde los guías podrían generar rutas turísticas a partir de la aplicación.

Además existen diferentes mejoras que podrían llevarse a cabo:

- Mejorar la interfaz gráfica: La aplicación tiene un aspecto simple y funcional. Por tanto, una persona experta en desarrollo de interfaces centradas en el usuario podría mejorar el aspecto visual de la misma de forma rápida y elegante.
- Incorporación de una plataforma social que permitiese compartir los recorridos de todos los usuarios. Ésta podría ser Facebook donde los usuarios podrían crear grupos para compartir las rutas generadas.
- Distinción de utilización de un profesor de Educación Física y un guía turístico e incorporar distintas funcionalidades según el tipo de usuario.

Capítulo 7

Bibliografía

[1] Federación española de deporte de orientación: <http://www.fedo.org>
(Última visita realizada a 17 de Septiembre de 2015)

[2] Página de la asociación PRAMES: <http://www.prames.com/>
(Última visita realizada a 17 de Septiembre de 2015)

[3] Página de la asociación Club IBÓN: <http://www.clubibon.es/>
(Última visita realizada a 17 de Septiembre de 2015)

[4] Página del grupo de investigación EFYPAF: <http://efypaf.unizar.es/>
(Última visita realizada a 17 de Septiembre de 2015)

[5] Página del grupo de investigación CeNITEQ:
<http://i3a.unizar.es/datos/grupo/ceniteq-109>
(Última visita realizada a 17 de Septiembre de 2015)

[6] Página oficial de Phonegap: <http://phonegap.com/>
(Última visita realizada a 17 de Septiembre de 2015)

[7] Página oficial de la herramienta Eclipse: <http://www.eclipse.org/>
(Última visita realizada a 17 de Septiembre de 2015)

[8] Página oficial de Google Developers:
<https://developers.google.com/?hl=es>
(Última visita realizada a 17 de Septiembre de 2015)